DEVELOPMENT OF ITERATIVE TECHNIQUES FOR THE SOLUTION OF UNSTEADY COMPRESSIBLE VISCOUS FLOWS

Grant. NAG-1-1217

Progress Report for the Period

August 14, 1991 - February 13, 1992

Submitted to

NASA Langley Research Center
Hampton, VA 23665

Attn: Dr. Woodrow Whitlow

Prepared by

Lakshmi N. Sankar
Professor, School of Aerospace Engineering

Duane Hixon
Graduate Research Assistant

School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332

February 1992

INTRODUCTION

This research project deals with the development of efficient iterative solution methods for the numerical solution of two- and three-dimensional compressible Navier-Stokes equations. The work during the present research period (August 14, 1991 - February 13, 1992) completes the two-dimensional applications, and begins the investigation of three-dimensional flow problems.

Iterative time marching methods have several advantages over classical multi-step explicit time marching schemes, and non-iterative implicit time marching schemes. Iterative schemes have better stability characteristics than non-iterative explicit and implicit schemes. Thus, the extra work required by iterative schemes per time step per node may usually be offset by the use of a larger time step. Iterative schemes can also be designed to perform efficiently on current and future generation scalable, massively parallel machines.

An obvious candidate for iteratively solving the system of coupled non-linear algebraic equations arising in CFD applications is the Newton method. Many investigators have implemented Newton's method in existing finite difference and finite volume methods. Depending on the complexity of the problem, the number of Newton iterations needed per step to solve the discretized system of equations can, however, vary dramatically from a few (3 to 5) to several hundred.

In this work, another popular approach based on the classical conjugate gradient method, known as the GMRES (Generalized Minimum Residual) algorithm is investigated . The GMRES algorithm has been used in the past by a number of researchers for solving steady viscous and inviscid flow problems with considerable success. Here, we investigate the suitability of this algorithm for solving the system of non-linear equations that arise in unsteady Navier-Stokes solvers at each time step.

Unlike the Newton's method which attempts to drive the error in the solution at each and every node down to zero, the GMRES algorithm only seeks to minimize the L2 norm of the error. In the GMRES algorithm the changes in

the flow properties from one time step to the next are assumed to be the sum of a set of orthogonal vectors. By choosing the number of vectors to a reasonably small value N (between 5 and 20) the work required for advancing the solution from one time step to the next may be kept to (N+1) times that of a non-iterative scheme. Many of the operations required by the GMRES algorithm such as matrix-vector multiplies, matrix additions and subtractions can all be vectorized and parallelized efficiently.

The dynamic stall of a NACA 0012 airfoil is the test case used to evaluate the various two dimensional time-accurate GMRES methods. The airfoil is pitched about the quarter chord point from 5 degrees to 25 degrees, at a reduced frequency of 0.151. The freestream Mach number is 0.283, and the Reynolds number is 3,450,000.

Progress During the Reporting Period

In January 1992, a paper concerning the two dimensional aspects of this work was presented at the Reno AIAA conference. A copy of that paper (AIAA Paper 92-0422) is enclosed with this report.

During the reporting period, the following tasks were completed:

a) Evaluation of 'Restart' GMRES for unsteady problems

A Newton iteration was added over the GMRES solver in order to reduce the number of directions (and hence, memory) needed for a given level of accuracy. Instead of a single 10 direction iteration at each time step, two five direction GMRES iterations were performed, with the first iteration providing an initial guess for the second. This cut the memory required for the GMRES routine in half, and the solution obtained was equal in accuracy to the single iteration computation. The only drawback is the increased CPU time necessary for the second GMRES matrix inversion.

Figure 1 shows the lift coefficient plotted as a function of time. The time step used is 20 times larger than the time step used in the original ADI non-iterative solver. It is seen that the five direction 'restart' GMRES (5:5/20) gives

almost identical results to the ten direction single iteration GMRES (10/20). Figure 2 shows the L2 norm of the global residual for these two computations. The 'restart' GMRES residual is much less 'noisy' than that of the single iteration. It is thought that this is due to the ability of the 'restart' solver to recover from a bad initial guess.

b) 'Dynamic Restart' GMRES solver

As Figure 2 shows, the residual of the restart solver varies with the nature of the flow field about the airfoil. When the flow is smooth and attached (on the upstroke), the residual is much lower than during the separated flow regime on the downstroke. It was noticed that the 5/20 single iteration solver gave identical lift and moment results to the 5:5/20 restart solver during the attached portion of the cycle. Therefore, an attempt was made to let the solver skip the second GMRES iteration if the residual was below a user-input value.

A value for the residual tolerance of 5×10^{-7} was tried, and this reduced CPU time by 30% from the previous 5:5/20 run. Figure 3 shows the lift coefficient results, and Figure 4 shows the global residuals.

c) Multigrid Steady and Unsteady Calculations

When a sample set of directions employed by the GMRES solver were plotted, it was seen that the initial directions are smooth (low frequency error), with the higher directions (above about five) becoming more and more jagged (high frequency error). Also, the initial directions are weighted much more heavily in the GMRES solution than the higher ones. In order to drive the low frequency errors to zero more rapidly, a multigrid 'Full Approximation Scheme (FAS)' was implemented.

The algorithm employed was a sawtooth pattern, with one level of coarse grid (fine-coarse-fine). With five directions in each iteration, this has the effect of putting a coarse grid evaluation into the 'restart' code.

Two steady calculations were made to validate the multigrid solver. The first was a transonic (M = 0.8), inviscid flow about a NACA 0012 airfoil at a 1.25 degree angle of attack. Figure 5 compares the 5 direction multigrid solver's global residuals to those of the original ADI code and a 40 direction fine grid

only GMRES solver. The multigrid solver is two times faster than the fine grid only GMRES solver, and requires 1/8 of the memory.

Figure 6 shows the results of a Navier-Stokes computation for the subsonic flow about a NACA 0012 airfoil at a five degree angle of attack. In this calculation, the freestream Mach number is 0.283, and the Reynolds number is 3,450,000. Again, a significant speedup is obtained while using a fraction of the memory.

At this point, the multigrid solver was implemented on the unsteady dynamic stall problem. Five directions were used, and the results compared to the results from the five direction restart solver. Since the same number of fine grid evaluations are performed, this shows the effect of the coarse grid evaluation on the solution. Results for the lift coefficient are given in Figure 7, and the global residual in Figure 8. It is seen that the residual is consistently lower only during the attached flow portion of the cycle, when the residual was already low. Since the multigrid solver didn't appear to have a positive effect on the residual during the separated flow portion of the cycle, it was felt that the CPU costs of the multigrid solver outweighed the benefits.

d) Improved formulation of the least squares matrix

At the end of the two dimensional work, an improved formulation of the least squares matrix was implemented in the GMRES routine. Details of the derivation are given in Appendix A. This formulation eliminates the dot products that were originally necessary to construct the least squares matrix, and reduced CPU time by 15% in a 10 direction GMRES calculation.

e) Three dimensional calculations

The GMRES solver was implemented on an existing 3-D Navier-Stokes wing code. An inviscid steady computation on a rectangular NACA 0012 wing was performed as an initial validation. Results for the global residual are plotted against the number of function evaluations required in Figure 9, and the lift coefficient history of this computation is given in Figure 10.

The GMRES solver is also being validated on steady and unsteady computations for the flow about an F-5 wing. Preliminary results have been obtained at this time.

CONCLUDING REMARKS

The two dimensional GMRES solver has provided a factor of two speedup for unsteady viscous dynamic stall calculations. An attempt at increasing accuracy by using a multigrid method in unsteady calculations was not very successful. Preliminary three-dimensional work has been performed, and initial results are encouraging.

Appendix A

Updated GMRES Formulation with New Least Squares Matrix

The J direction vectors are found as follows:

First, the initial direction is computed as

$$\vec{d}_1 = \mathbf{M}(q^{n+1,k}) \tag{A1}$$

and normalized as

$$\vec{d}_1 = \frac{\vec{d}_1}{\left\|\vec{d}_1\right\|} \tag{A2}$$

To compute the remaining search directions (j=1,2,..,J-1), take

$$\vec{d}_{j+1} = \overline{\mathbf{M}}(q^{n+1,k}\,;\,\vec{d}_j) - \sum_{i=1}^{j} b_{ij}\,\vec{d}_i \tag{A3}$$

where

$$b_{ij} = (\overline{\mathbf{M}}(q^{n+1,k}\,;\,\vec{d}_j),\,\vec{d}_i) \tag{A4}$$

and

$$\overline{\mathbf{M}}(q\,;\,\vec{d}) = \frac{\mathbf{M}(q + \varepsilon\vec{d}) - \mathbf{M}(q)}{\varepsilon}. \tag{A5}$$

Here, e is taken to be some small number. In this work, e is taken to be 0.001.

The new direction $\vec{d}_{j+1}$ is normalized before the next direction is computed:

$$b_{j+1,j} = \left\|\vec{d}_{j+1}\right\|, \tag{A6}$$

and

$$\vec{d}_{j+1} = \frac{\vec{d}_{j+1}}{b_{j+1,j}}. \tag{A7}$$

After obtaining the search directions, the solution vector is updated using

$$\mathbf{q}^{n+1,k+1} = \mathbf{q}^{n+1,k} + \sum_{j=1}^{J} a_j \vec{d}_j \qquad \text{(A8)}$$

where the coefficients a_j are chosen to minimize:

$$\left\| \mathbf{M}(\mathbf{q}^{n+1,k+1}) \right\|^2 = \left\| \mathbf{M}(\mathbf{q}^{n+1,k} + \sum_{j=1}^{J} a_j \vec{d}_j) \right\|^2$$

$$\cong \left\| \mathbf{M}(\mathbf{q}^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{\mathbf{M}}(\mathbf{q}^{n+1,k}; \vec{d}_j) \right\|^2 . \qquad \text{(A9)}$$

This equation is minimized as follows:

Let D_j be the matrix of directions $\{d_1, d_2, d_3, ..., d_j\}$. Also, let F_j be the matrix of directional derivatives given as $\{M_1, M_2, M_3, ..., M_j\}$, where:

$$M_j = \overline{\mathbf{M}}(\mathbf{q}^{n+1,k}; \vec{d}_j) . \qquad \text{(A10)}$$

Then Eq. (A3) may be rewritten in matrix form as:

$$M_j = D_{j+1} B \qquad \text{(A11)}$$

Here, B is the (J+1) x (J) matrix:

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & & & b_{1,J-2} & b_{1,J-1} & b_{1,J} \\ b_{2,1} & b_{2,2} & b_{2,3} & & \cdots & b_{2,J-2} & b_{2,J-1} & b_{2,J} \\ 0 & b_{3,2} & b_{3,3} & & & b_{3,J-2} & b_{3,J-1} & b_{3,J} \\ & 0 & & & \ddots & & \vdots & \\ & & & & & b_{J-1,J-2} & b_{J-1,J-1} & b_{J-1,J} \\ & 0 & & & 0 & 0 & b_{J,J-1} & b_{J,J} \\ & & & & & 0 & 0 & b_{J+1,J} \end{bmatrix} \qquad \text{(A12)}$$

Note that at this point, $b_{J+1,J}$ is not yet known. Saad and Schultz give the following formula for evaluating this term without another function evaluation:

$$b_{J+1,J}^2 = \left\| \overline{\mathbf{M}}(\mathbf{q}^{n+1,k}; \vec{d}_J) \right\|^2 - \sum_{i=1}^{j} b_{i,J}^{\ 2} \qquad \text{(A13)}$$

At this point, Eq. (A9) is rewritten:

$$\left\| \mathbf{M}(\mathbf{q}^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{\mathbf{M}}(\mathbf{q}^{n+1,k}; \vec{d}_j) \right\|^2 = \left\| \mathbf{M}(\mathbf{q}^{n+1,k}) + \mathbf{M}_j \mathbf{A} \right\|^2 \qquad \text{(A14)}$$

where A is the vector $\{a_1, a_2, a_3, ..., a_j\}^T$. Then, using the definition of the first direction and Eq. (A11), Eq. (A14) becomes:

$$\begin{aligned} &\left\| \mathbf{M}(\mathbf{q}^{n+1,k}) + \mathbf{M}_j \mathbf{A} \right\|^2 \\ &= \left\| \left(\left\| \vec{d}_1 \right\| \vec{d}_1 + \mathbf{M}_j \mathbf{A} \right) \right\|^2 \\ &= \left\| \left(\left\| \vec{d}_1 \right\| \vec{d}_1 + \mathbf{D}_{j+1} \mathbf{B} \mathbf{A} \right) \right\|^2 \\ &= \left\| \mathbf{D}_{j+1} \left(\left\| \vec{d}_1 \right\| \mathbf{e} + \mathbf{B} \mathbf{A} \right) \right\|^2 \\ &= \left\| \left(\left\| \vec{d}_1 \right\| \mathbf{e} + \mathbf{B} \mathbf{A} \right) \right\|^2 \end{aligned} \qquad \text{(3.23)}$$

where e is the first column of the (JxJ) identity matrix.

This least squares problem is solved using the QR algorithm in LINPACK.

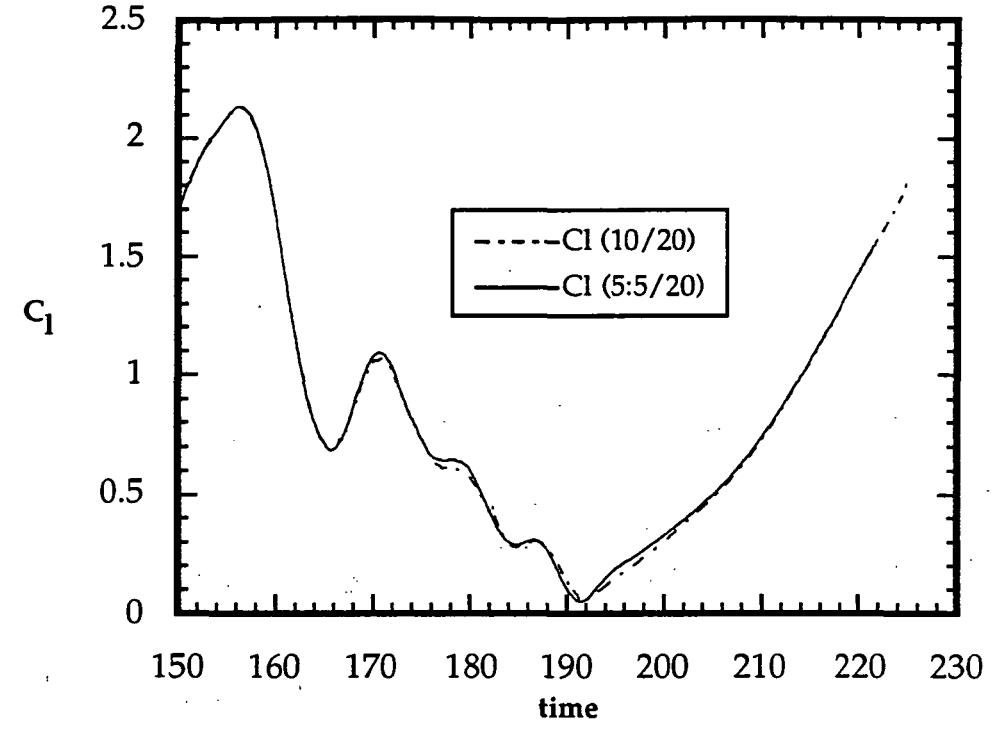

Figure 1: Restart GMRES and Single Iteration GMRES Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k = 0.151; Re = 3,450,000)

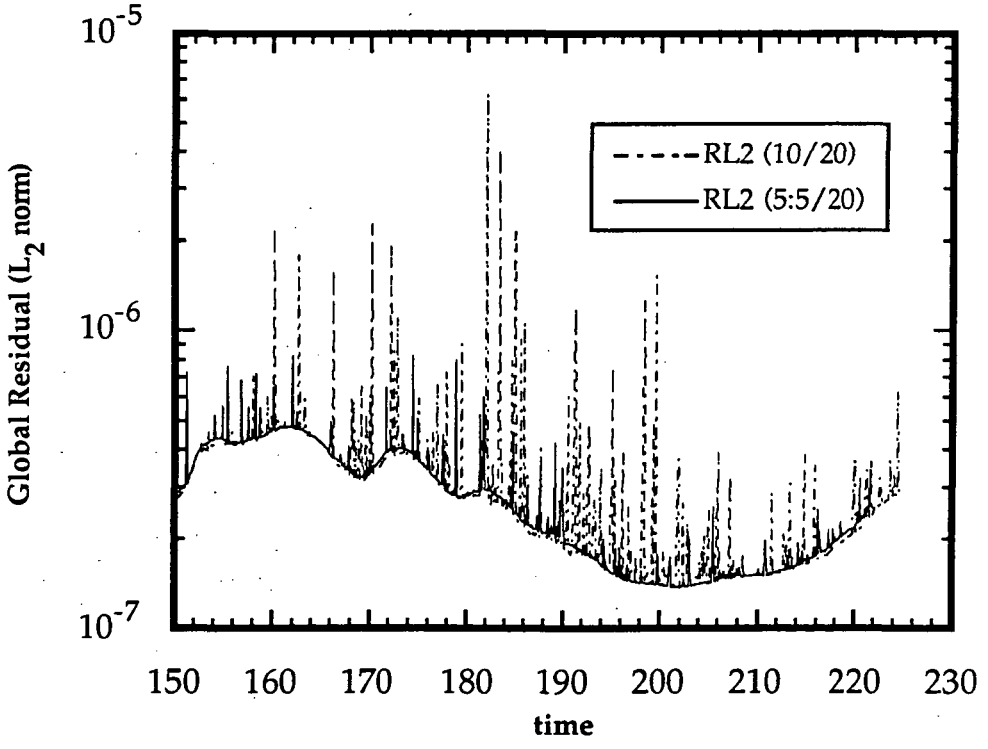

Figure 2: Restart GMRES and Single Iteration GMRES Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k = 0.151; Re = 3,450,000)

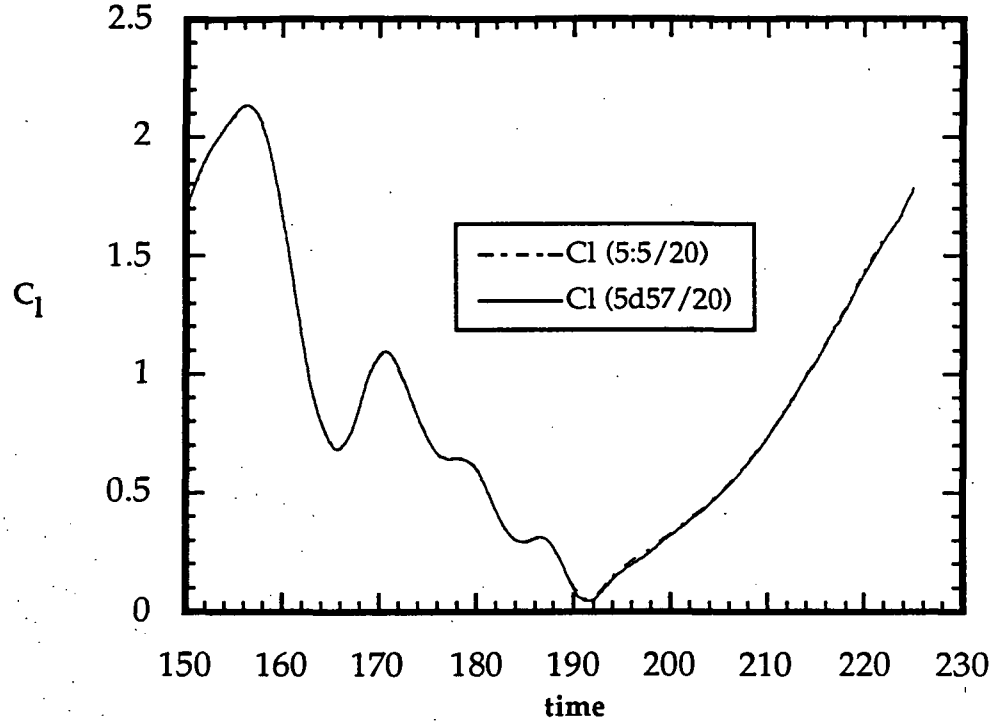

Figure 3: Dynamic Restart GMRES and Restart GMRES Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k = 0.151; Re = 3,450,000)

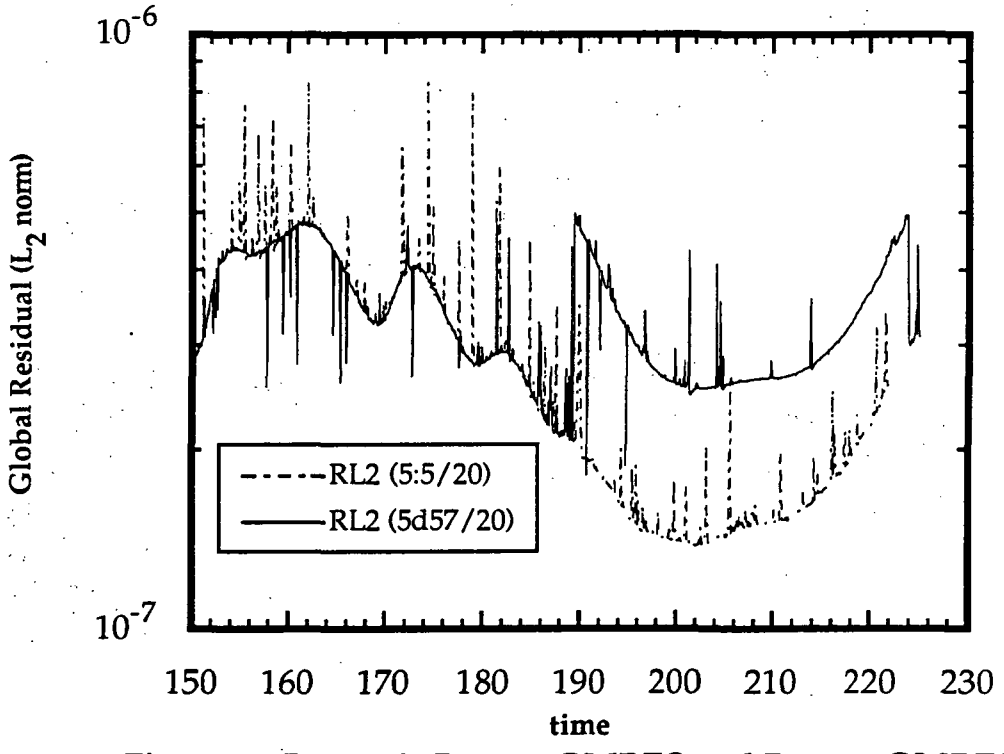

Figure 4: Dynamic Restart GMRES and Restart GMRES Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k=0.151; Re = 3,450,000)

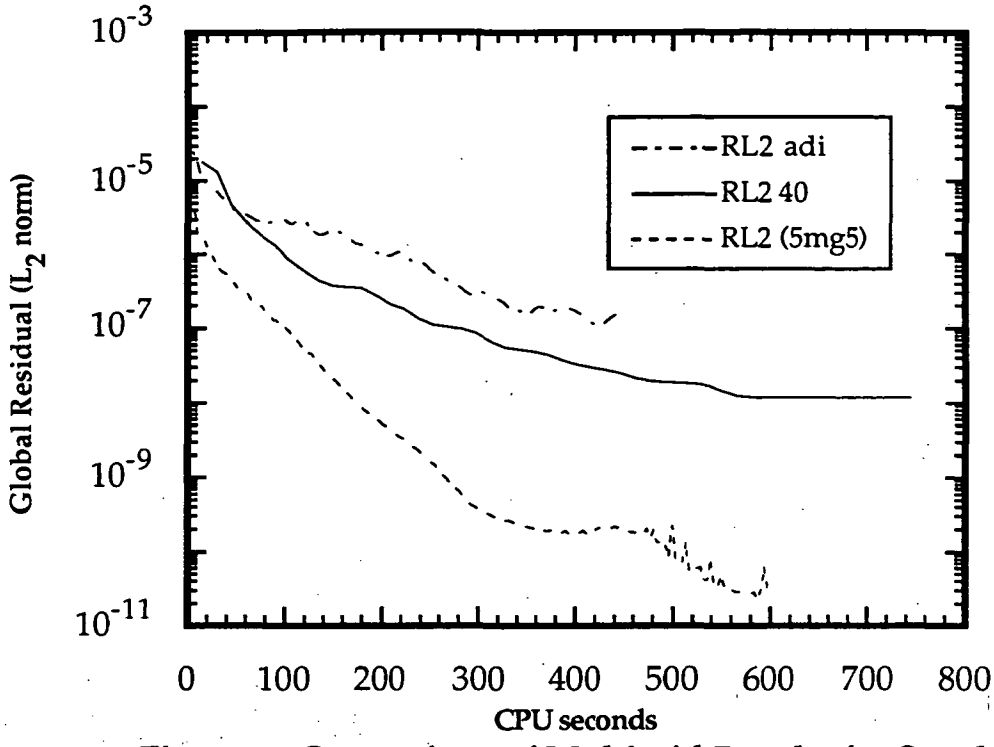

Figure 5: Comparison of Multigrid Results for Steady InviscidTransonic Calculation

(M = 0.8; α = 1.25 deg)

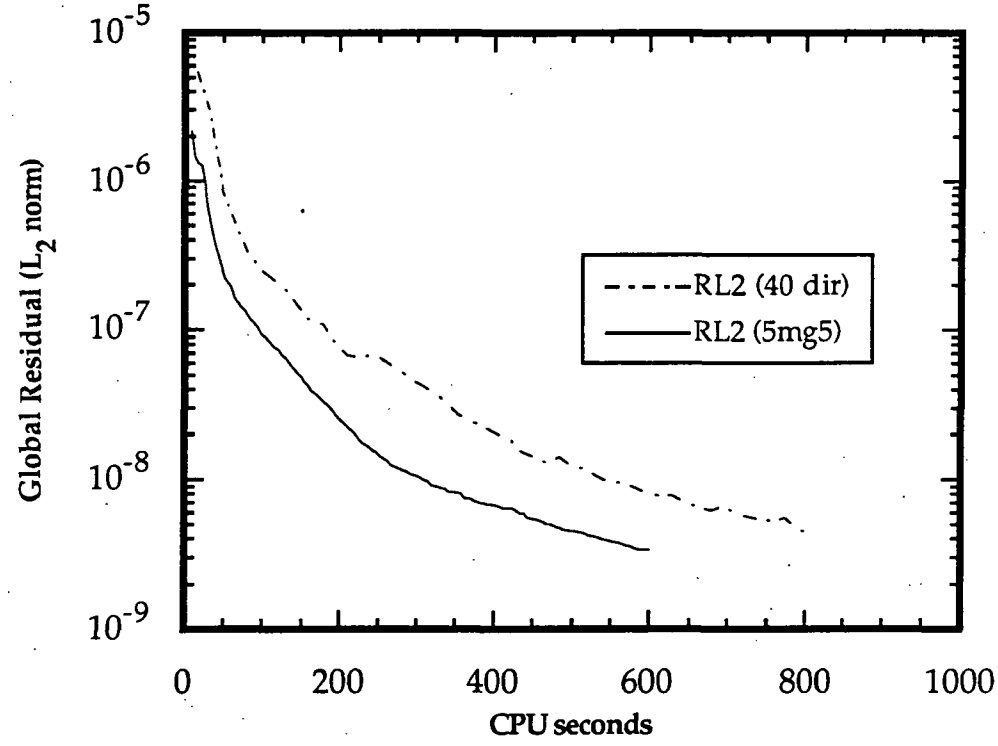

Figure 6: Comparison of Multigrid Results for Steady Navier-Stokes Calculation

(M = 0.283; α = 5 deg.; Re = 3,450,000)

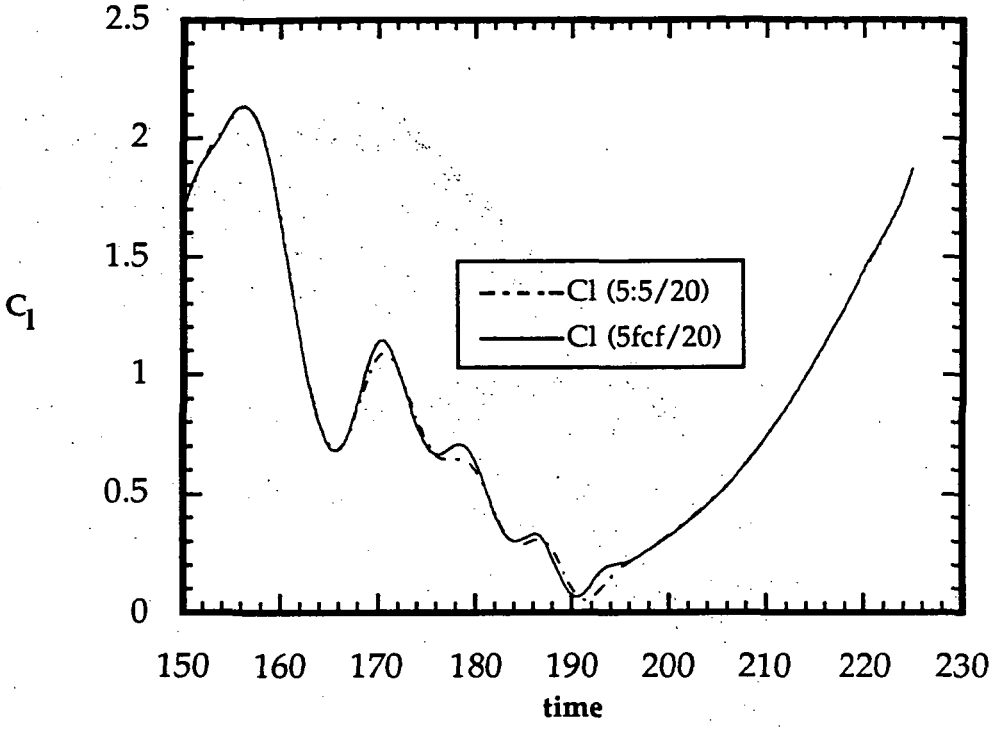

Figure 7: Comparison of Unsteady Multigrid Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k = 0.151; Re = 3,450,000)

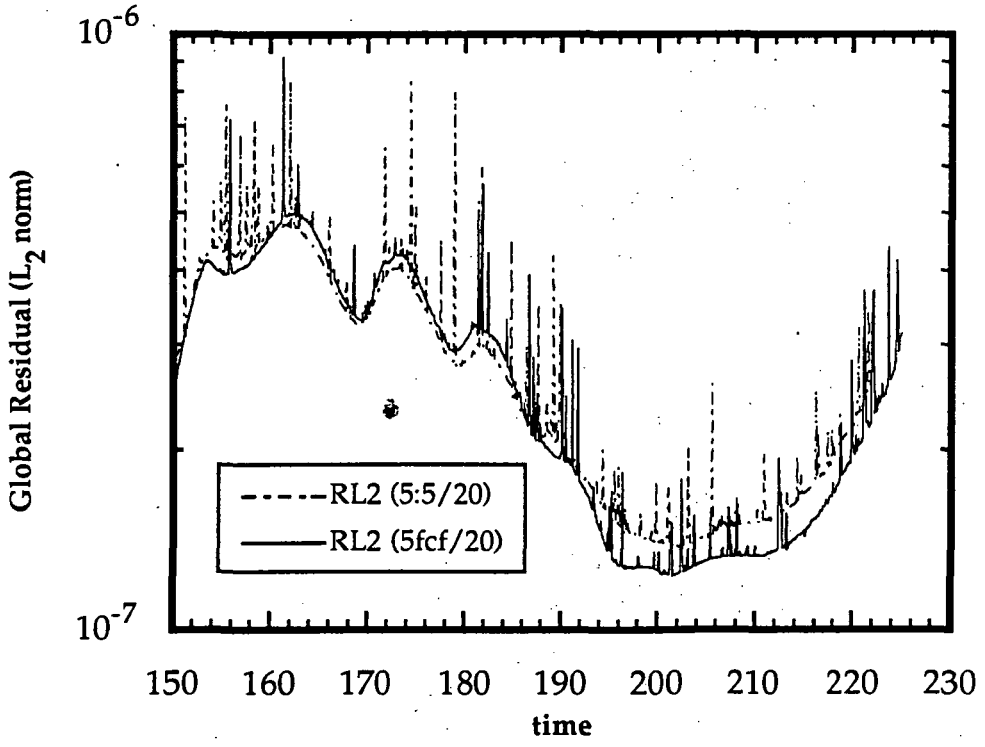

Figure 8: Comparison of Unsteady Multigrid Results for a NACA 0012 Airfoil in Dynamic Stall (M = 0.283; k = 0.151; Re = 3,450,000)

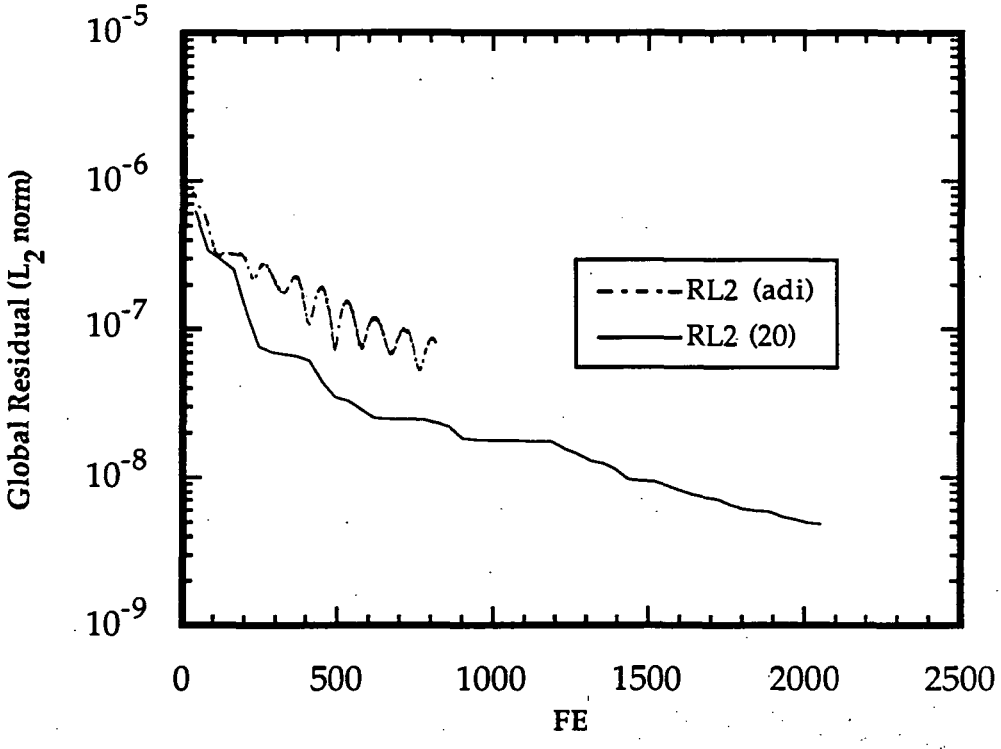

Figure 9: GMRES Euler Calculation for a 3-D Steady NACA 0012 Wing (M = 0.120; α = 8 deg.; AR = 5)

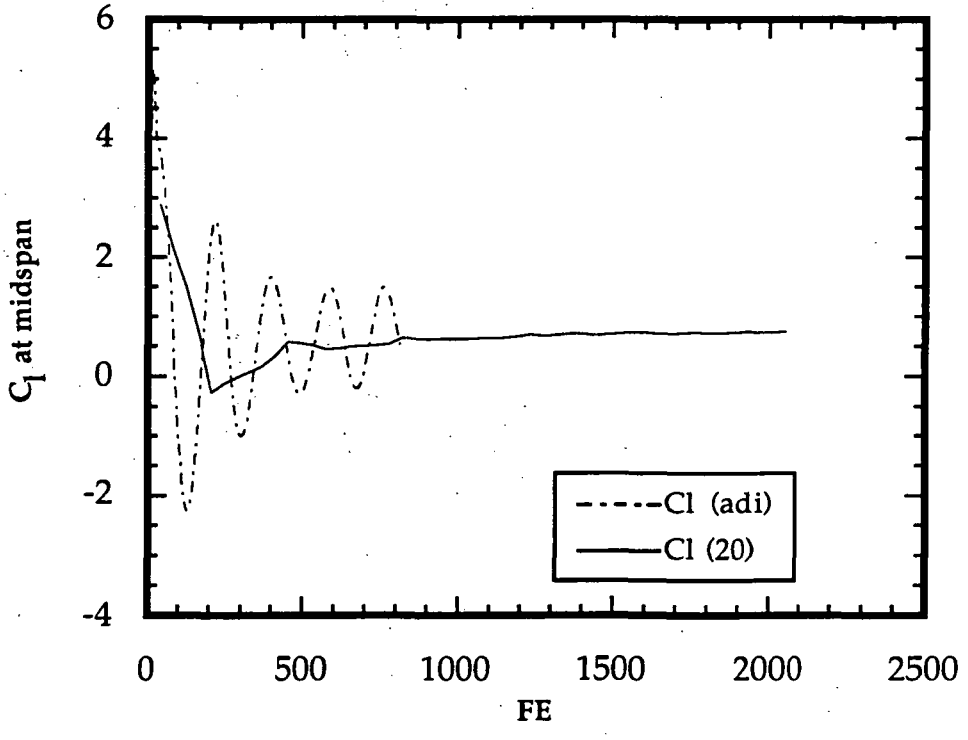

Figure 10: GMRES Euler Calculation for a 3-D Steady NACA 0012 Wing (M = 0.120; α = 8 deg.; AR = 5)

AIAA 92-0422
Application of a Generalized Minimal Residual Method to 2D Unsteady Flows

R. Hixon and L.N. Sankar
Georgia Tech
Atlanta, GA

30th Aerospace Sciences Meeting & Exhibit
January 6-9, 1992 / Reno, NV

APPLICATION OF A GENERALIZED MINIMAL RESIDUAL METHOD TO 2D UNSTEADY FLOWS

Ray Hixon* and L.N. Sankar**
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Abstract

A generalized minimum residual scheme (GMRES), previously developed for solving nonlinear and linear systems of equations, has been applied to the numerical solution of 2-D unsteady compressible flows. It is found that the use of GMRES significantly increases the time step that may be used, compared to non-iterative implicit schemes. The feasibility of reducing the memory requirements of the GMRES scheme using a multigrid strategy has also been explored. Several sample steady and unsteady viscous flow applications are presented.

Introduction

During the past two decades, there has been significant progress in the field of numerical simulation of unsteady compressible viscous flows. At present, a variety of solution techniques exist such as the transonic small disturbance analyses (TSD) [1,2,3], transonic full potential equation-based methods[4,5,6], unsteady Euler solvers[7,8], and unsteady Navier-Stokes solvers[9,10,11,12]. These advances have been made possible by developments in three areas: (1) Improved numerical algorithms, (2) Automation of body-fitted grid generation schemes, and (3) Advanced computer architectures with vector processing and massively parallel processing features.

Despite these advances, numerical simulation of unsteady viscous flows still remains a computationally intensive problem, even in two dimensions. For example, the problem of dynamic stall of an oscillating NACA 0012 airfoil using state of the art alternating direction implicit (ADI) procedures presently require between 10,000 and 20,000 time steps per cycle of oscillation at low reduced frequencies when the viscous flow region is sufficiently resolved[9]. In three dimensions, unsteady Navier-Stokes simulations of a helicopter rotor blade in forward flight requires over 30,000 time steps or more for a full revolution of the rotor[10]. In other unsteady flows, such as the high angle of attack flow past fighter aircraft configurations, a systematic parametric study of the flow is presently not practical due to the very large CPU time needed for the simulations[13]. Thus, it is clear that significant improvements to the existing algorithms, or dramatic improvements in computer architectures will be needed, before unsteady viscous flow analyses become practical day-to-day engineering tools.

One scheme that has been of recent interest is the Generalized Minimal RESidual (GMRES) method originally proposed by Saad and Schultz[14]. This procedure uses a conjugate gradient-like method to accelerate the convergence of existing flow solvers. GMRES was added to existing steady flow solvers by Wigton, Yu, and Young[15], and to an unstructured grid flow solver by Venkatakrishnan and Mavriplis[16]. Saad has also used a Krylov subspace projection method on a steady, incompressible Navier-Stokes problem and an unsteady one dimensional wave propagation equation[17]. To our knowledge, GMRES has not been applied to multi-dimensional unsteady compressible flow problems.

In this paper, the GMRES scheme has been considered as a candidate for acceleration of a Newton iteration time marching scheme for unsteady 2-D compressible viscous flow calculation; this has provided significant reductions in the computer time requirements over the existing class of explicit and implicit

---

* Graduate Research Assistant, School of Aerospace Engineering. Member of AIAA.

** Professor , School of Aerospace Engineering. Senior Member AIAA.

time marching schemes. The proposed method has been tested on structured grids, but is flexible enough for extension to unstructured grids. The described scheme has been tested only on the current generation of vector processor architectures of the Cray Y/MP class, but should be suitable for adaptation to massively parallel machines.

Mathematical and Numerical Formulation

Underlying Newton Based Formulation

A starting point for the GMRES method is an existing flow solver. The Newton iteration time marching scheme has been used for the 2-D compressible Navier-Stokes equations on a curvilinear coordinate system. The Newton scheme and the combined Newton/GMRES scheme is , however, applicable to 3-D flows on curvilinear body-fitted coordinate systems.

The governing equations may be written formally as:

$$q_t + F_x + G_y = R_x + S_y \qquad (1)$$

Here **q** is the vector containing the flow properties such as density, u- and v-momentum per unit volume, and total energy per unit volume. The terms F and G represent the transport of mass, momentum, and energy by convection, and also include pressure effects. The terms R and S represent viscous stress effects, heat conduction, and the friction-generated heat.

For simplicity, the algorithm is described for the Cartesian form shown above (Eq. (1)).

The objective of the calculation is to determine **q** at a time level 'n+1' given the values of **q** at a previous time level 'n'. On a stretched Cartesian grid, at a typical node (i,j), this equation may be discretized as:

$$\frac{(q_{i,j}^{n+1} - q_{i,j}^{n})}{\Delta t} + \delta_x F^{n+m} + \delta_y G^{n+m} = \delta_x R^{n+m} + \delta_y S^{n+m} \qquad (2)$$

The above discretization is first order accurate in time if 'm' is set to zero or one, and second order accurate if 'm' is set to 1/2. The operators δ_x and δ_y represent second order accurate or fourth order accurate spatial differences. The terms F and G are numerical fluxes that differ from the physical fluxes F and G in that they incorporate artificial viscosity terms, or changes to F and G needed to make the scheme upwinded. In the present studies, which primarily deal with subsonic and transonic applications, the numerical viscosity model proposed by Jameson, Turkel, and Schmidt and modified by Swanson and Turkel is used[15].

In the past, equation set (2) was solved by non-iterative time marching schemes [10].

A variant of the non-iterative time marching schemes is an iterative time marching scheme. Several researchers have used Newton-iteration schemes in steady and unsteady Navier-Stokes calculations[16]. In this approach, a sequence of sub-iterations (k = 0,1,2,...) are used within each time step. Equation (2) is rewritten as follows:

$$\frac{(q_{i,j}^{n+1,k} - q_{i,j}^{n})}{\Delta t} + \delta_x F^{n+m,k} + \delta_y G^{n+m,k} = \delta_x R^{n+m,k} + \delta_y S^{n+m,k} \qquad (3)$$

The terms F, G, R, and S at time-iteration level (n+m,k) are expanded about their values at the time level 'n+m' and at the previous iteration level 'k-1'. This leads to a system of coupled, linear equations for the changes in q between two successive iterations:

$$[M]\{\Delta q\} = \{R\} \qquad (4)$$

where

$$\Delta q = q^{n+1,k} - q^{n+1,k-1} \qquad (5)$$

and {R} is the residual:

$$\{R\} = -\frac{(q_{i,j}^{n+1,k-1} - q_{i,j}^{n})}{\Delta t} - \delta_x F^{n+m,k-1} - \delta_y G^{n+m,k-1} + \delta_x R^{n+m,k-1} + \delta_y S^{n+m,k-1} \qquad (6)$$

The objective of the Newton iteration scheme is to solve equation set (3) by repeated application of equation set (4). The matrix [M] is a banded 5- or 9- diagonal matrix whose individual elements are 4x4 matrices. This matrix is usually approximately factored into

tri-diagonal matrices and inverted. Equation set (4) is solved until the residual R is driven to zero. In a full Newton iteration scheme, the elements of the coefficient matrix will be recomputed every iteration, based on $q^{n+1,k-1}$. When {R} approaches zero, equation (2) is exactly satisfied.

The advantage of a Newton iteration scheme, particularly in the context of approximate factorization schemes, is that the errors associated with the factorization method can be reduced or removed. That is, as Δq goes to zero, the errors associated with the approximate factorization of [M] do not affect the solution. By specifying a convergence criteria for Δq, one can also ensure that equation set (2) is satisfied at every time step to within a user-specified tolerance. The disadvantage of the above type of Newton iteration schemes is that each Newton iteration requires approximately the same amount of CPU time as a single step using a non-iterative time marching scheme. To be cost-effective, a Newton-iteration based scheme that uses, say, 5 iterations per time step should use a CFL number that is, on the average, 5 times larger than the CFL number associated with a non-iterative scheme.

GMRES Formulation

The Newton formulation given above may be expressed in this way:

$$q^{n+1,k+1} = F(q^{n+1,k}) \qquad (7)$$

In words, given a guess for $q^{n+1,k}$, the Newton solver returns a (hopefully) better approximation $q^{n+1,k+1}$ to the correct solution. When the solution has converged (i.e., $q^{n+1,k} = q^{n+1,k+1}$), then:

$$q^{n+1,k} - F(q^{n+1,k}) = M(q^{n+1,k}) = 0 \qquad (8)$$

The GMRES solver uses the original Newton solver as a function evaluator (i.e., given a set of input flow properties, the Newton solver sends back an updated set of flow properties), and computes the set of flow properties that will satisfy Eq. (8) at each time step.

It should be noted that the GMRES scheme only uses the original flow solver as a 'black box' to determine the effect of changing the input flow properties on the residual M. Because of this, the GMRES solver is very portable, and can easily be implemented in a wide variety of codes regardless of the original code's solution procedure (as long as a residual for Eq. (8) can be defined). This is a major advantage of the GMRES acceleration method over schemes which are tied closely to the details of the algorithm (e.g., multigrid methods).

Let Δq be the change in q between successive Newton iterations (i.e., $q^{n+1,k+1} - q^{n+1,k}$).

The GMRES solver starts by assuming that the Δq required to set the residual given by Eq. (8) to zero lies in the vector space made of a set of orthonormal direction vectors. In a two dimensional flow problem, there are a total of 4 x imax x kmax possible direction vectors (i.e., changing one variable at one point is a direction; changing another variable at the same point is another direction orthogonal to the first.). In a 'J' direction GMRES iteration, the (4 x imax x kmax) dimension space of orthogonal direction vectors is collapsed down to a (J) dimension space. In this problem, this results in computing a J dimensional space instead of a 25,748 dimensional space (for imax = 157 and kmax = 41 and J < 20).

Once the directions are defined, the slope of the residual in each direction is calculated by moving a small distance in this direction from the starting point and solving for the residual vector, then subtracting the result from the residual vector from the starting point and dividing by the distance. From here, a least squares problem is solved to reduce the residual as much as possible by using a linear combination of the directions.

Obviously, the success and speed of the GMRES solution method depends greatly on the original flow solver's ability to help define useful direction vectors, and hence a subspace that contains many of the important error components.

Closely following the development and notation given by Wigton, Yu, and Young[15], the J direction vectors are found as follows:

First, the initial direction is computed as

$$\vec{d}_1 = M(q^{n+1,k}) \qquad (9)$$

and normalized as

$$\vec{d}_1 = \frac{\vec{d}_1}{\left\| \vec{d}_1 \right\|} \qquad (10)$$

where $\| d \|$ is the dot product of the vector d with itself.

To compute the remaining search directions (j=1,2,..,J-1), take

$$\vec{d}_{j+1} = \overline{M}(q^{n+1,k} ; \vec{d}_j) - \sum_{i=1}^{j} b_{ij} \vec{d}_i \qquad (11)$$

where

$$b_{ij} = (\overline{M}(q^{n+1,k} ; \vec{d}_j), \vec{d}_i) \qquad (12)$$

and the derivative of the error in the jth direction is given as

$$\overline{M}(q^{n+1,k} ; \vec{d}_j) = \frac{M(q^{n+1,k} + \varepsilon \vec{d}_j) - M(q^{n+1,k})}{\varepsilon} . \qquad (13)$$

Here, ε is taken to be some small number, and (b,d) is the dot product of the vectors b and d. In this problem, $\varepsilon = .001$ was found to give good results, following a range of ε values attempted: $.00001 < \varepsilon < .1$.

The new direction $\vec{d}_{j+1}$ is normalized before the next direction is computed:

$$\vec{d}_{j+1} = \frac{\vec{d}_{j+1}}{\left\| \vec{d}_{j+1} \right\|} . \qquad (14)$$

After obtaining the search directions, the solution vector is updated using

$$q^{n+1,k+1} = q^{n+1,k} + \sum_{j=1}^{J} a_j \vec{d}_j \qquad (15)$$

where the coefficients a_j are chosen to minimize:

$$\| M(q^{n+1,k+1}) \|^2 \cong \left\| M(q^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{M}(q^{n+1,k} ; \vec{d}_j) \right\|^2 \qquad (16)$$

This least squares problem is solved using QR reduction., as discussed in the appendix.

The work per time step is approximately equal to J+1 times a single Newton iteration, where J is the number of direction vectors used. Beyond this, there is a JxJ matrix inversion at each GMRES step, but this doesn't appreciably affect the time for J<20. Thus, compared to a non-iterative ADI scheme, this method is (J+1) times more expensive.

Of course, the objective of using GMRES in this manner is to lower the overall computation time for a given unsteady problem. By using the present approach, the time step only has to be small enough to capture the physics of the flow (in other methods, e.g. non-iterative ADI schemes, the time step had to be small enough to keep factorization errors relatively small). The hope is that the number of GMRES directions necessary for a given level of accuracy will be significantly less than the larger time step that is allowed by making the procedure iterative (e.g., 10 directions of GMRES, each requiring one ADI step, with 20 times the original time step is roughly a 2x speedup).

Results

All of the calculations presented here were done on an algebraic 157 x 41 grid. All CPU times are from the NASA-Langley Cray Y/MP.

Validation of GMRES code

Two cases were run with GMRES to validate it against the original Newton code, applied as a non-iterative ADI solver.

The first case was inviscid transonic flow (Mach number of 0.8) over a NACA 0012 airfoil at a 1.25 degree angle of attack.This problem was chosen to see the effects of shocks on the GMRES solver. Figures 1 and 2 give the residual and lift coefficient history comparisons between the original ADI solver and the GMRES (40) code. The GMRES (40) solver requires only 50-55% of the CPU time necessary for the ADI code to reach a given

level of convergence. Also, the lift coefficient converges much more rapidly.

The interesting part of this problem was in choosing the number of GMRES directions to use. When less than 40 directions were employed, the residual would drop very quickly; then stall out and not decrease. A run of 80 directions showed that there was a limit to the speedup obtainable from using more directions. It is thought that the higher directions contain much more noise than the early ones, and thus degrade the solution. Figure 3 shows a comparison of the global residuals for various GMRES runs.

One case was run with GMRES to validate it in the Navier-Stokes mode. The problem calculated was that of a NACA 0012 airfoil at a 5 degree angle of attack at M = 0.283 and a Reynolds number of 3,450,000.

Two GMRES runs were performed, with 10 and 40 directions used. Residual and lift coefficient histories are given in Figures 4 and 5, and the pressure distribution is compared to the ADI result in Figure 6. Excellent agreement is shown between the solvers. Also, as in the inviscid case, an increase in the number of directions allows a further reduction in the L_2 norm of the residual.

Unsteady Flow Analyses

Once the code was validated, several preliminary 2-D unsteady calculations were performed using the GMRES solver to determine if significant savings in CPU time may be obtained compared to the original ADI scheme.

In the following discussion, the term 'residual' refers to the left hand side of Eq. (8). This is a measure of the accuracy to which the discretized equation (RHS of Eq. (6)) is satisfied.

The first test case evaluates the solver's ability to handle unsteady transonic flow. A plunging NACA 64A010 airfoil at a Mach number (M_∞) of 0.8 and a reduced frequency based on half chord of 0.2 is solved in the Euler mode. The plunging motion is defined by the equation

$$Y_t = -M_\infty \sin(1^\circ)\sin(\omega t) \qquad (17)$$

At first, a time step of 20 times the ADI time step was employed, but it became apparent that this was too large to resolve the shock motion properly. A time step factor of 5 was found to be small enough to adequately resolve the physics of the problem, but the GMRES was not stable using less than 10 directions (100% increase in computer time). This illustrates the tradeoff between having the large time step necessary for effective speedup with GMRES and the small enough time step to accurately model the physics of the problem. This may be peculiar to inviscid flows where a relatively coarse grid will allow large time steps.

The lift and pitching moment histories are plotted as a function of phase angle, ωt, and are compared with the Euler calculations by Steger[19] in Figures 7 and 8.

Another case which was tested is a Navier-Stokes calculation for a NACA 0012 airfoil in the deep dynamic stall condition. The Mach number is 0.283, the Reynolds number is 3.45 million, and the reduced frequency is 0.151. The airfoil motion is defined by

$$\alpha = 15^\circ - 10^\circ \cos(\omega t) \qquad (18)$$

A time step factor of 20 was tried initially. To get a comparison with the original ADI code, 20 directions were run (20/20). Note that this takes slightly longer than the original ADI code to run, mainly due to the matrix inversion during the GMRES calculation. Figures 9, 10, and 11 compare the GMRES results with experimental results by McAlister et al[20]. While the GMRES (20/20) code does not get quantitatively good results, the result follows the experiments qualitatively. Thus, the GMRES (20/20) run was chosen as a benchmark to compare later runs to. Figure 12 gives the residual history of the GMRES (20/20) run.

The next series of runs were performed to see what sort of speedups were likely from GMRES. For this set, a time step of 20 times the ADI time step was used (i.e., GMRES (x/20)). The number of directions were set at 10 and 5. Results for lift, moment, and residual are shown in Figures 13, 14, and 15. These are plotted against time as it is easier to judge results in this way. The output shows that GMRES (10/20) is very nearly as good as

(20/20), while accuracy falls off in the (5/20) run.

The last series of runs were done to see the effect of the time step on the GMRES solver. From the results of the last series, GMRES (x/2x) was chosen (number of directions equal to half of the time step factor). These results are shown in Figures 16, 17, 18, 19, 20, and 21. The results were split into two groups to keep the graphs legible. From these graphs, it can be seen that there is a tradeoff between accuracy of the GMRES iteration (which goes up with number of directions) and the time step necessary to resolve flow phenomena. From this series of runs, it appears that a time factor of 20 is the best choice in this case.

Another experiment was tried to reduce the amount of memory required for the GMRES calculation. In this run, two Newton iterations per time step were done, and GMRES was applied during each Newton iteration (e.g., two 5 direction GMRES iterations instead of one 10 direction iteration per time step). The advantage was that the memory necessary for the GMRES iteration was cut in half.

It was found that the 'restart' method worked better than the single step method for this case. The residual had much less 'noise' than before, and was lower. Figure 22 compares the residual histories of the two runs, while Fig. 23 shows the lift coefficient histories.

It was noticed that the number of directions needed for a given level of convergence was less in the portion of the cycle where the flow is attached. To take advantage of this, a switching mechanism based on residual was implemented in the restart solver. In this variant, the second GMRES iteration is not performed if the residual is below a user-specified tolerance. This resulted in a 30% speedup over the original restart code when a tolerance of 5×10^{-7} was input. Results of this run are given in Figures 24 and 25. Net speedup over the original ADI solver was a factor of 2.0 (3173 CPU seconds from 6200).

Multigrid Analysis

At this point, a multigrid solver was introduced to try to reduce the number of GMRES directions necessary for convergence (and thus reduce the total memory required). In each iteration, the variables are transferred to a coarse grid and a GMRES iteration is performed there. It was postulated that this coarse grid calculation would be able to capture low frequency components of the correction vector, while the fine grid captured the high frequency components. The multigrid solver used three 5 direction GMRES iterations per time step in a fine-coarse-fine sawtooth pattern. In order to compare these with prior results, it was decided to use the same number of fine grid directions per iteration.

To validate the multigrid solver, the same steady runs were performed. It is seen in Fig. 26 and 27 that the multigrid solver gives impressive speedups as compared to the fine-grid-only GMRES results. One noticeable difference was that the transonic steady case only took 5 directions to converge (down from 40 with only the fine grid).

The multigrid solver was then run in unsteady mode on the dynamic stall test case. In Figures 28, 29, and 30, a (20/20) run is compared to a fine-grid-only (5:5/20) run (two 5 direction Newton iterations per time step) and a F-C-F (5:5/20) run (a 5 direction evaluation on the fine grid, then the coarse grid, then on the fine grid again). In effect, this is testing the effectiveness of the coarse grid evaluation. As seen in Fig. 30, no appreciable gain due to multigrid (i.e., order of magnitude reduction in the residual) is apparent except when the flow is attached and the flowfield is relatively smooth.

Concluding Remarks

The possibility of accelerating 2-D unsteady compressible flow calculations using a GMRES method has been investigated. A multigrid version of the code has also been evaluated. Encouraging results have been obtained. The solver is now being expanded to three dimensions.

Acknowledgements

This work was supported by a grant from NASA Langley Research Center (Grant No. NAG-1-1217). Dr. John B. Malone was the technical monitor.

References

1. Borland, C.J. and Rizzetta, D., "Nonlinear Transonic Flutter Analysis," AIAA Paper 81-

0608-CP, AIAA Dynamic Specialists Conference, 1981.

2. Rizzetta, D.P. and Borland, C., "Numerical Solution of Unsteady Transonic Flow over Wings with Viscous-Inviscid Interaction", AIAA Paper 82-0352, January 1982.

3. Batina, J. T., "Unsteady Transonic Algorithm Improvements for Realistic Aircraft Applications", AIAA Paper 88-0105, January 1988.

4. Sankar, L.N., Malone, J.B., and Tassa, Y., "An Implicit Conservative Algorithm for Steady and Unsteady Three-Dimensional Transonic Potential Flows", AIAA Paper 81-1016-CP, June 1981.

5. Malone, J.B. and Sankar, L.N., "Application of a Three-Dimensional Steady and Unsteady Full Potential Method for Transonic Flow Computations", AFWAL-TR-84-3011, Flight Dynamics Laboratory, Wright Patterson Air Force Base, Dayton, Ohio, 1984.

6. Shankar, V., Ide, H., Gorski, J. and Osher, S., "A Fast, Time-Accurate Unsteady Full Potential Scheme", AIAA Paper 85-1512-CP, July 1985.

7. Pulliam, T.H., and Steger, J.L., "Implicit Finite Difference Simulations of Three-Dimensional Compressible Flow", AIAA Journal, Vol. 18, 1980.

8. Batina, J.T., "Unsteady Euler Solutions Using Unstructured Dynamic Meshes", AIAA Paper No. 89-0115, January 1989.

9. Sankar, L. N. and Tang, W., "Numerical Solution of Unsteady Viscous Flow past Rotor Sections", AIAA Paper 85-0129.

10. Wake, B. E. and Sankar, L. N., "Solution of the Navier-Stokes Equations for the Flow About a Rotor Blade", Journal of the American Helicopter Society, April 1989.

11. Rai, M. M., "Navier-Stokes Simulations of Rotor-Stator Interaction Using Patched and Overlaid Grids", AIAA Paper 85-1519-CP, July 1985.

12. Gatlin, B. and Whitfield, D. L., "An Implicit Upwind Finite Volume Scheme for Solving the Three-Dimensional Thin-Layer Navier-Stokes Equations", AIAA Paper 87-1149-CP, June 1987.

13. Sankar, L. N. and Kwon, O. J., "Viscous Flow Simulation of Fighter Aircraft", AIAA Paper 91-0278, January 1991.

14. Saad, Y. and Schultz, M.H, "GMRES: A Gnerealized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems", SIAM J. Sci. Stat. Comp., Vol. 7, No. 3, 1986, pp.856-869.

15. Wigton, L. B., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamics Codes", AIAA Paper 85-1494-CP, 1985.

16. Venkatakrishnan, V. and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes", ICASE Report 91-40, May 1991.

17. Saad, Y. and Semeraro, B. D. , :Application of Krylov Exponential Propagation to Fluid Dynamics Equations", AIAA Paper 91-1567-CP, 1991.

18. Swanson, R. C. and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations", AIAA Paper 87-1107-CP, June 1987.

19. Steger, J. L., "Implicit Finite Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 18, No. 2, pp. 159-167, Feb. 1980.

20. McAlister, K.W., Pucci, S.L., McCroskey, W.J., and Carr, L.W., "An Experimental Study of Dynamic Stall on Advanced Airfoil Section, Volume 2: Pressure and Force Data", NASA TM 84245, Sept. 1982.

Appendix A

The GMRES procedure assumes that the correction vector Δq required to solve Eq. (8) lies in a 'J' dimensional subspace of the entire problem. Thus, the correction vector has the form:

$$\Delta q = \sum_{j=1}^{J} a_j \vec{d}_j \qquad \text{(A1)}$$

where the d_j's are unit vectors in orthogonal directions.

Once these unit vectors are defined in the subspace (which is the first part of the GMRES algorithm), and the derivatives of the residual M in these directions are calculated, the correction vector is then computed using:

$$M(q^{n+1,k+1}) \cong M(q^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{M}(q^{n+1,k}; \vec{d}_j) = 0 \quad \text{(A2)}$$

This equation is underdetermined, as the M vectors are 'L' long(where L = imax x jmax x 4), while there are only 'J' coefficients. Eq. (A2) is solved in the following way:

Eq. (A2) may be rewritten:

$$[X]\{a\} = -\{b\} \quad \text{(A3)}$$

where [X] is the residual derivative matrix:

$$[X] = \begin{bmatrix} (\overline{M}(q;\vec{d}_1))_1 & \cdots & (\overline{M}(q;\vec{d}_J))_1 \\ \vdots & \ddots & \vdots \\ (\overline{M}(q;\vec{d}_1))_L & \cdots & (\overline{M}(q;\vec{d}_J))_L \end{bmatrix} \quad \text{(A4)}$$

and {a} is the coefficient vector. The right hand side is given as:

$$\{b\} = \begin{Bmatrix} (M)_1 \\ \vdots \\ (M)_L \end{Bmatrix} \quad \text{(A5)}$$

To solve this problem, Eq. (A3) is multiplied by the transpose of the [X] matrix:

$$[X]^T[X]\{a\} = -[X]^T\{b\} \quad \text{(A6)}$$

The left hand side now is a symmetric J x J matrix:

$$[X]^T[X] = \begin{bmatrix} (\overline{M}_1, \overline{M}_1) & \cdots & ((\overline{M}_1, \overline{M}_J)) \\ \vdots & \ddots & \vdots \\ (\overline{M}_J, \overline{M}_1) & \cdots & ((\overline{M}_J, \overline{M}_J)) \end{bmatrix} \quad \text{(A7)}$$

where

$$\overline{M}_j = \overline{M}(q;\vec{d}_j) \quad \text{(A8)}$$

The right hand side of Eq. (A6) becomes:

$$\{b\} = \begin{Bmatrix} ((\overline{M}_1, M)) \\ \vdots \\ ((\overline{M}_J, M)) \end{Bmatrix} \quad \text{(A9)}$$

Eq. (A6) is then solved by QR reduction. This is important if the matrix $[X]^T[X]$ is not well conditioned.

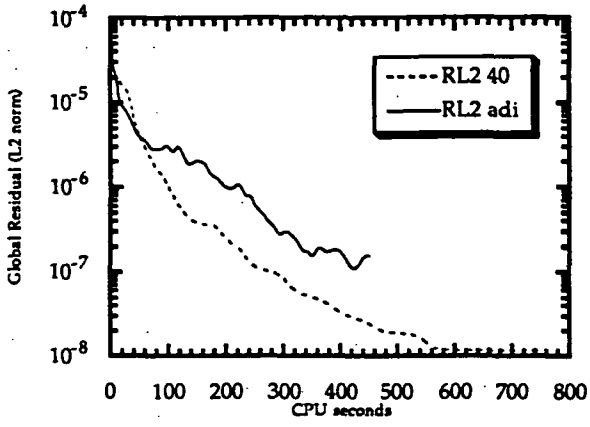

Figure 1
Residual History for a NACA 0012 Airfoil
(M=0.8; α = 1.25 deg)

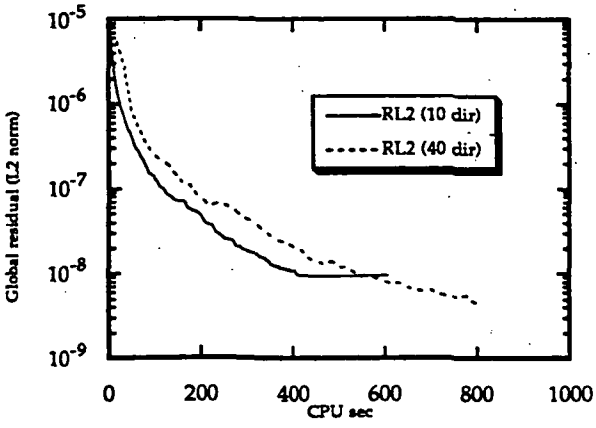

Figure 4
Residual History for a NACA 0012 Airfoil
(M=0.283; α=5 deg; Re=3,450,000)

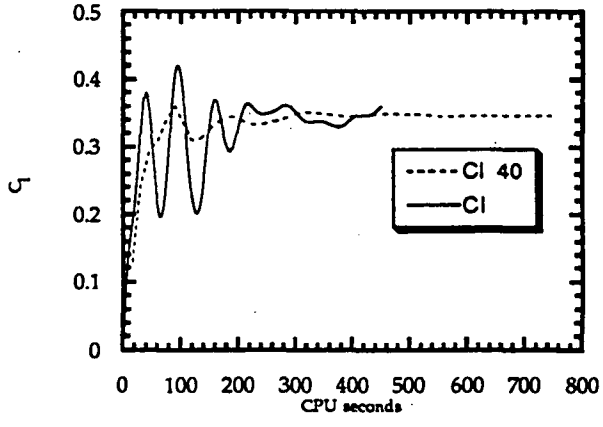

Figure 2
Lift Coefficient Histories for a NACA 0012 Airfoil
(M = 0.8; α = 1.25 deg)

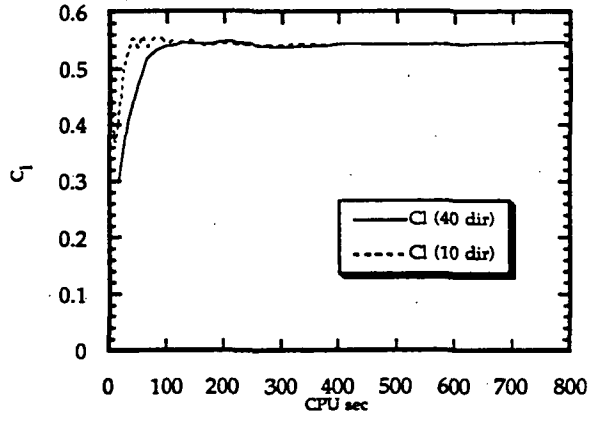

Figure 5
Cl Histories for a NACA 0012 Airfoil
(M = 0.283; α = 5 deg.; Re=3,450,000)

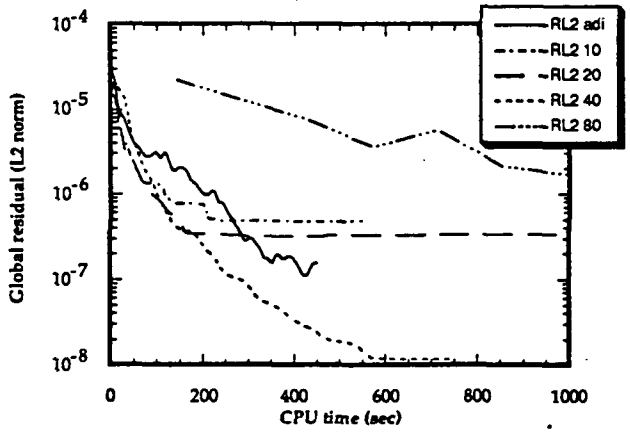

Figure 3: Comparison of Residual for
Steady Transonic Inviscid Calculation
(NACA 0012, M = 0.8, α = 1.25 deg)

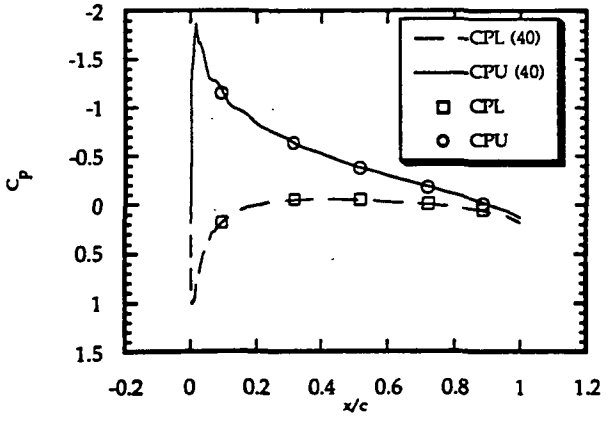

Figure 6
C_p distribution about a NACA 0012 Airfoil
(M = 0.283; α = 5 deg.; Re=3,450,000)

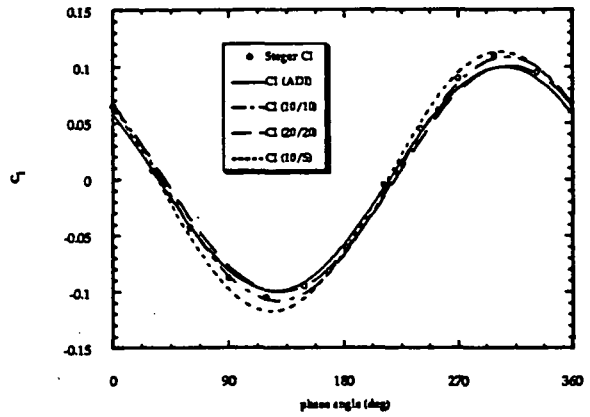

Figure 7: Effect of Time Step on GMRES Result for Lift Coefficient of a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

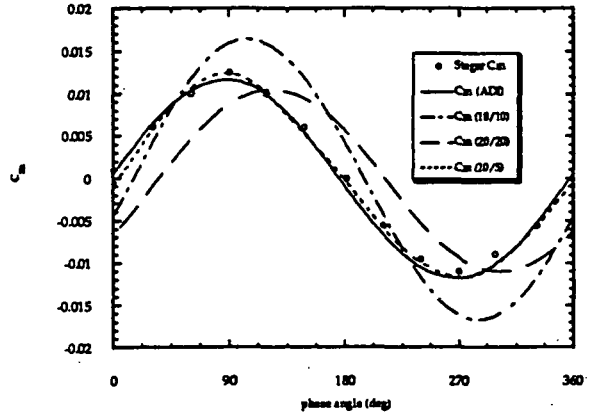

Figure 8: Effect of Time Step on GMRES Result for a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

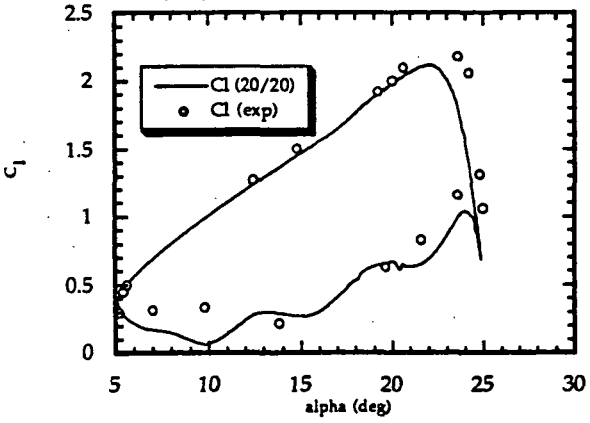

Figure 9: Comparison of GMRES (20/20) with Experimental Results for Lift Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151)

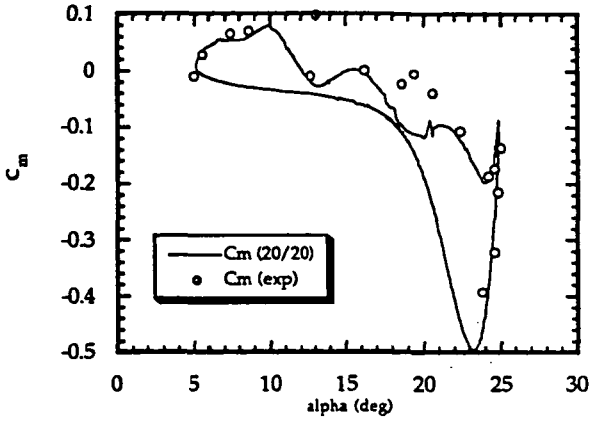

Figure 10: Comparison of GMRES (20/20) with Experimental Data for Coefficient of Moment (NACA 0012; M = 0.283; k = 0.151)

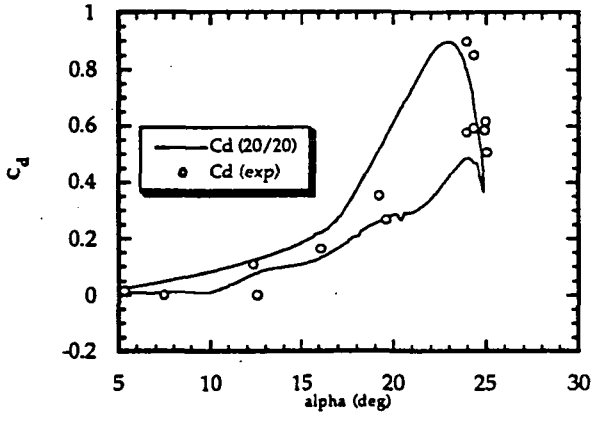

Figure 11: Comparison of GMRES(20/20) with Experimental Results for Drag Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151)

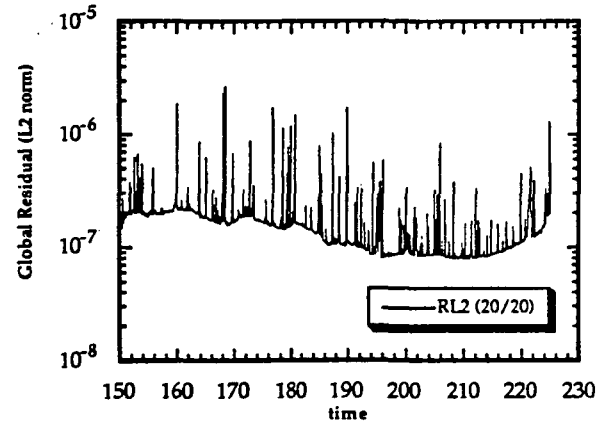

Figure 12: GMRES (20/20) L2 Residual results for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

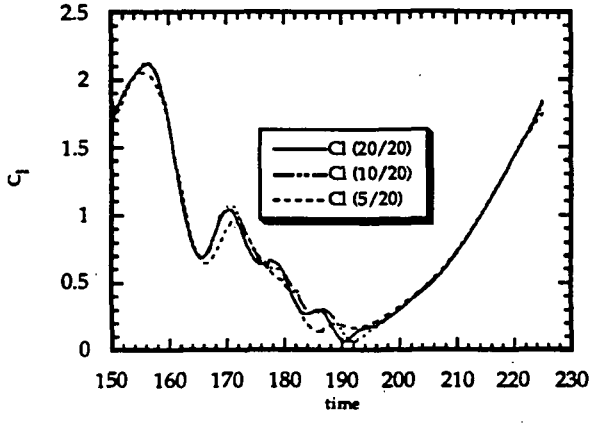

Figure 13: Effect of Directions on GMRES (x/20) Results for Lift Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k= 0.151; Re = 3,450,000)

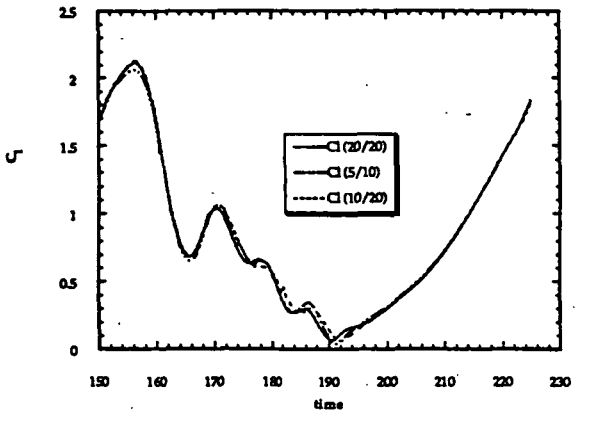

Figure 16: Comparison of GMRES (x/2x) results with GMRES (20/20) for Lift of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

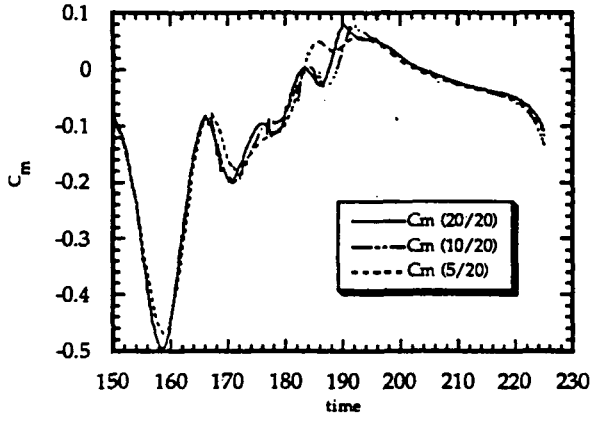

Figure 14: Effect of Directions on GMRES (x/20) Results for Moment Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

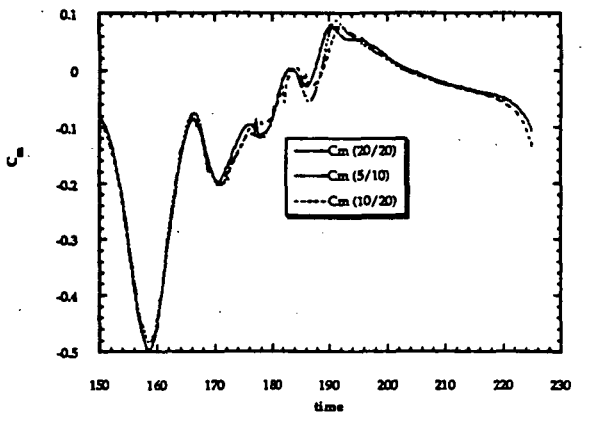

Figure 17: Comparison of GMRES (x/2x) Results with GMRES (20/20) Results for Moment Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

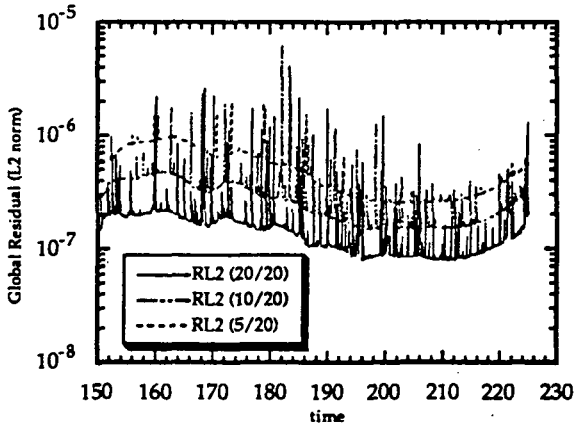

Figure 15: Effect of Directions on Residual of GMRES (x/20) for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

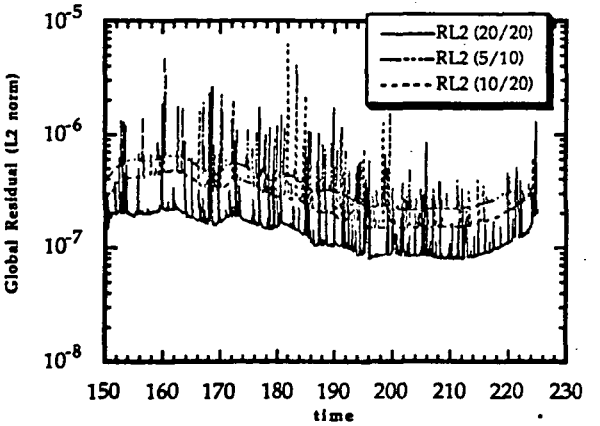

Figure 18: Residual History of the GMRES (x/2x) solvers Compared with GMRES (20/20) for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

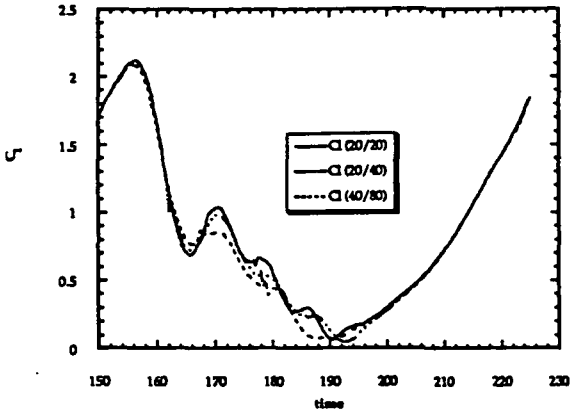

Figure 19: Comparison of GMRES (x/2x) Results with GMRES (20/20) for Lift Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k=0.151; Re = 3,450,000)

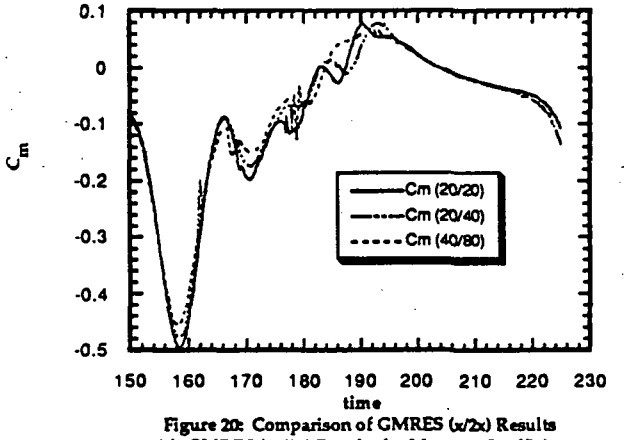

Figure 20: Comparison of GMRES (x/2x) Results with GMRES (20/20) Results for Moment Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

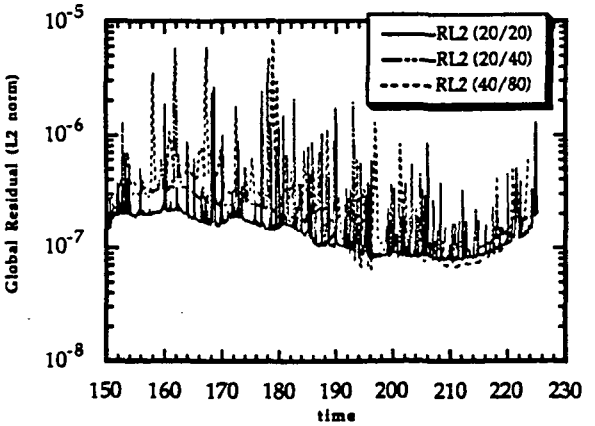

Figure 21: Comparison of GMRES (x/2x) with GMRES (20/20) Residual for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

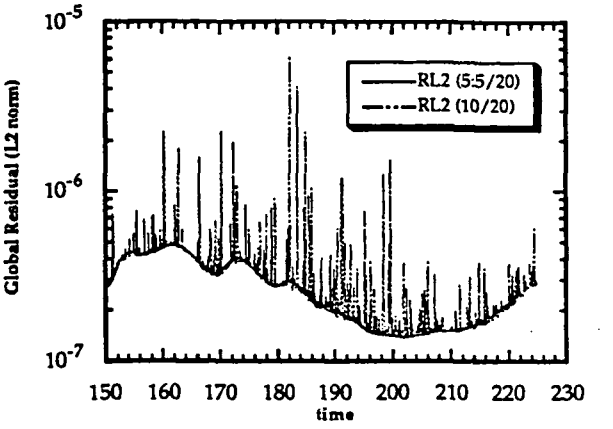

Figure 22: Comparison of Restarted GMRES (5:5/20) with GMRES (10/20) Residual for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

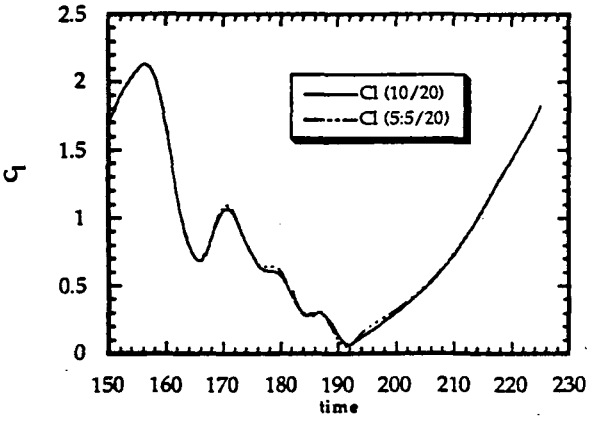

Figure 23: Comparison of Restarted GMRES (5:5/20) and GMRES (10/20) Lift Coefficients for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

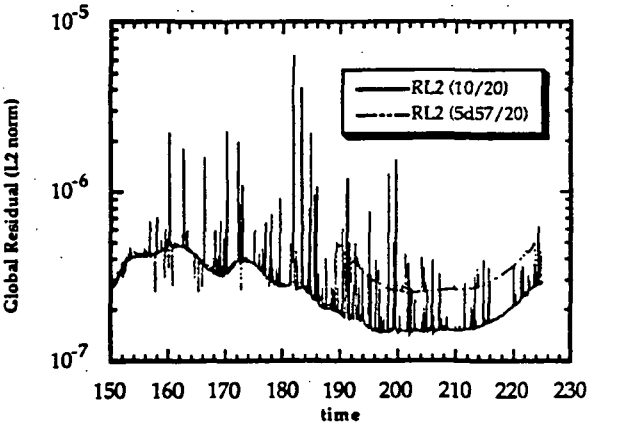

Figure 24: Comparison of Dynamic Restart GMRES with GMRES (10/20) Residual for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

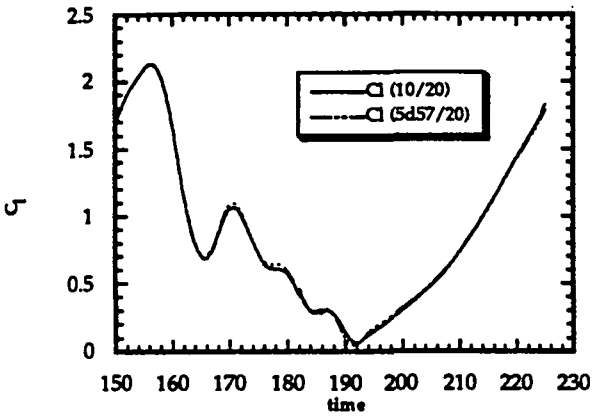

Figure 25: Comparison of Dynamic Restart GMRES with GMRES (10/20) Lift Coefficient for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

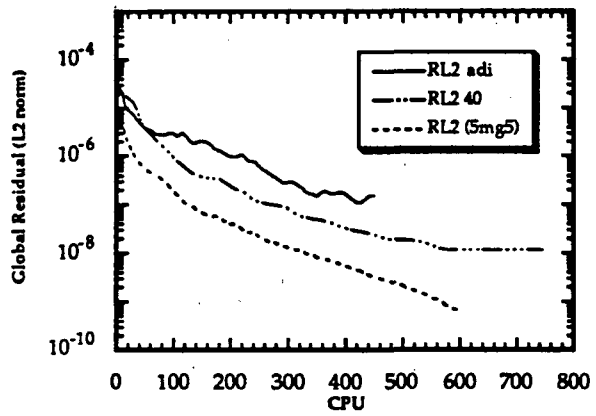

Figure 26: Convergence of Inviacid Transonic Steady Case (NACA 0012; M = 0.8; α = 1.25 deg.)

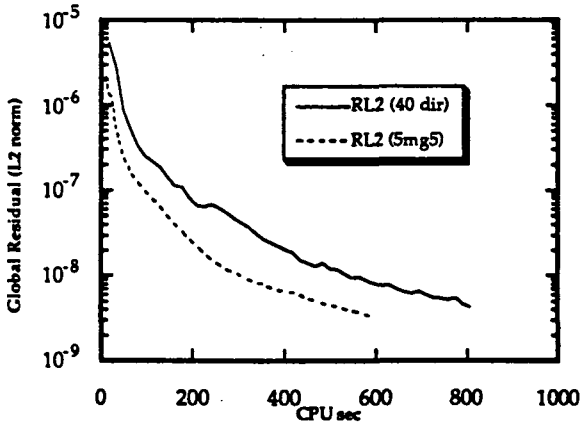

Figure 27: Comparison of Multigrid GMRES and GMRES (40) Residual Histories for a Steady NACA 0012 Airfoil (M = 0.283; α = 5 deg.; Re = 3,450,000)

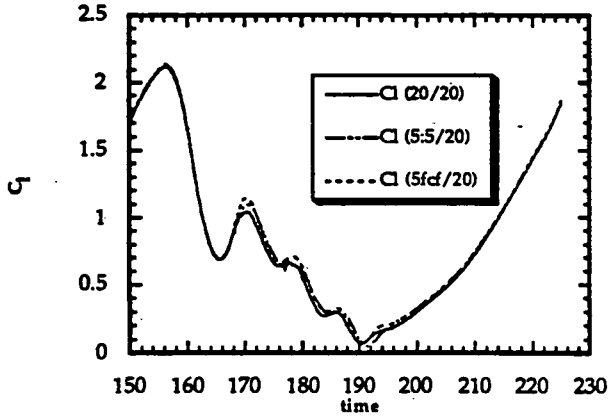

Figure 28: Comparison of Lift Coefficients for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

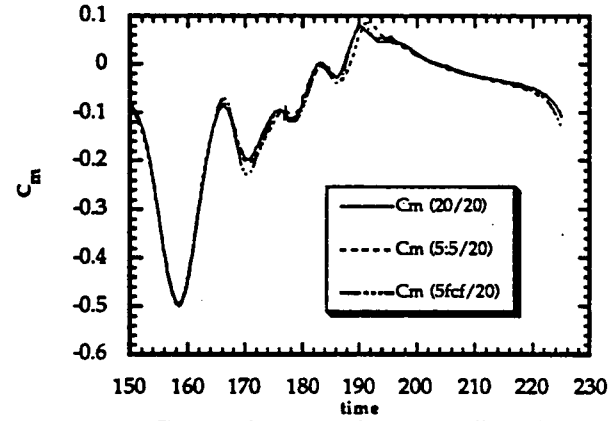

Figure 29: Comparison of Moment Coefficient for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

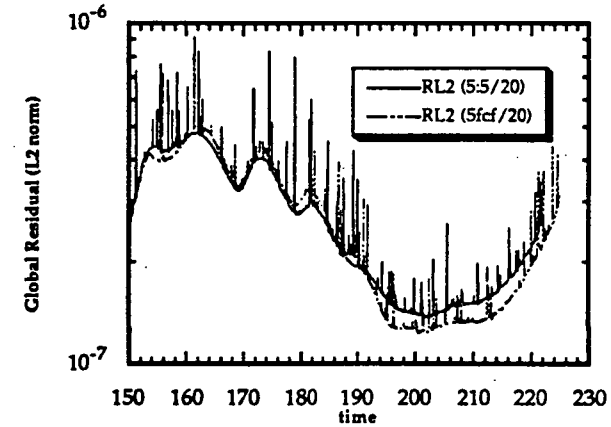

Figure 30: Comparison of Global Residuals for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)